UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/811,177	03/26/2004	James F. Riordan	CH920020047US1	2029

48233 7590 03/26/2008

SCULLY, SCOTT, MURPHY & PRESSER, P.C.
400 GARDEN CITY PLAZA
SUITE 300
GARDEN CITY, NY 11530

EXAMINER	
TRUVAN, LEYNNA THANH	

ART UNIT	PAPER NUMBER
2135	

MAIL DATE	DELIVERY MODE
03/26/2008	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/811,177	RIORDAN, JAMES F.	
	Examiner	Art Unit	
	Leynna T. Truvan	2135	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *28 December 2007*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *3-5,8-11 and 13-20* is/are pending in the application.

4a) Of the above claim(s) *1-2, 6-7, 12, and 21-22* is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *3-5,8-11 and 13-20* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☒ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☒ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. Claims 3-5, 8-11, and 13-20 are pending.

Claims 1-2, 6-7, 12, and 21-22 are cancelled by applicant.

Claim Objections

2. ***Claims 3 and 8 are objected to because of the following informalities:***

On line 14 of claim 1, recites "determining that the system in corrupted if ...new secret", should be "is corrupted".

On line 17 of claim 1, recites "determining that the system in corrupted if ...new secret", should be "is corrupted".

Appropriate correction is required.

Response to Arguments

3. ***Applicant's arguments with respect to claims 3-5, 8-11, and 13-20 have been considered but are moot in view of the new ground(s) of rejection.***

Examiner traverses that Ogg's RSA algorithm does not employ a hash function. Ogg does not only teach RSA encryption, but DES or DES MAC cryptography that involves a secret, cryptographic function, and hash function (col.18, lines 40-44 and col.22, lines 20-40). The claimed invention broadly recites secret, cryptographic function and evolved secret. Thus, the RSA method is not the only cryptography method to read on the broad claims, but other cryptography (i.e. DES, DES MAC, DSA or a signature) can also suggest the claimed invention.

Ogg shows a table where the RSA engine is capable of performing the following modular arithmetic and exponentiation functions for high speed RSA encryption and including a signature (col.8, lines 10-40). Ogg discloses encryption and authentication of the key token (secret) format: DES, DSA, and RSA (col.18, lines 40-44 and col.22, lines 20-40). Ogg explains the HMAC is a digital signature created using a hash algorithm with an arbitrary message and the secret key (col.32, lines 45-65). Thus, Ogg obviously includes a cryptographic function comprising a hash function and an exponentiation function. A secondary prior art is brought forth to further clarify obviousness to the entirety of the claimed invention.

Graunke discloses the present invention binds the integrity of a given application to its ability to perform some cryptographic operation using an asymmetric key pair in a manner that is tamper resistant. The goal is to prevent an unencrypted copy of digital content to be made. Graunke includes integrity verification kernels (IVKs), the use of an asymmetric key pair and a key compiler, and tamper resistance methods. It combines the cryptographic technologies of digital signatures and certificates with tamper resistant software to improve the integrity that is very difficult to modify without detection (col.4, lines 8-22). Thus, Graunke's binding a secret to data involves digital signatures and certificates as similar to Ogg's invention. This obviously improves the integrity making it very difficult to modify without detection. Further, Graunke discloses the IVK (Integrity Verification Kernel) Gen function creates an IVK source code module that uses the asymmetric public key as the root of trust (col.9, lines 25-30). The key compiler generates IVK source code for key module for calculating digital signatures and the source code computing a cryptographic hash function followed by modular exponentiation (col.9, lines 32-41). Therefore, it would have been obvious for a person of ordinary skills in the art to combine

the teachings of Ogg with Graunke to teach cryptographic function comprises a hash function and the hash function comprises an exponentiation function because to verify the data is validated and trusted for use in tamper detection of content (Graunke – col.9, lines 38-41 and col.10, lines 30-39).

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

4. *Claims* 3-5, 8-11, and 13-20 *are rejected under 35 U.S.C. 103(a) as being unpatentable over Ogg, et al. (US 7,236,956), and further in view of* Graunke, et al. *(5,991,399).*

As per claim 3:

Ogg discloses a method for detecting an attack on a data processing system, the method comprising, in the data processing system:

providing an initial secret; (col.22, lines 48 and col.23, lines 46-48)

[binding] the initial secret (col.19, lines 6-18 and col.21, lines 7-14) to data indicative of an initial state of the system via a cryptographic function; (col.23, lines 57-67 and col.24, lines 45-52)

recording state changing administrative actions performed on the system in a log; (col.30, lines 44-47 and col.39, lines 50-52)

prior to performing each state changing administrative action, generating a new secret by performing the cryptographic function on a combination of data indicative of the administrative action and the previous secret (col.19, lines 44-48 and col.20, lines 39-52), and erasing the previous secret; (col.24, lines 58-67)

evolving the initial secret based on the log to produce an evolved secret; comparing the evolved secret with the new secret; (col.33, lines14-19)

determining that the system is uncorrupted if the comparison indicates a match between the evolved secret and the new secret; and (col.33, lines 21-25)

determining that the system in corrupted if the comparison indicates a mismatch between the evolved secret and the new secret, (col.33, lines 38-55)

[*wherein the cryptographic function comprises a one-way hash function and the hash function comprises an exponentiation function*]. (col.8, lines 10-40 and col.32, lines 45-65)

Ogg suggests private key used by modules to decrypt client secrets transmitted to the module during registration (col.23, lines 57-67 and col.33, lines 15-22), which seems to suggest a form of binding an initial secret to data indicative of an initial state of the system via a cryptographic function (col.19, lines 6-18 and col.21, lines 7-14). Ogg shows a table where the RSA engine is capable of performing the following modular arithmetic and exponentiation

functions for high speed RSA encryption and including a signature (col.8, lines 10-40). Ogg discloses encryption and authentication of the key token (secret) format: DES, DSA, and RSA (col.18, lines 40-44 and col.22, lines 20-40). Ogg explains the HMAC is a digital signature created using a hash algorithm with an arbitrary message and the secret key (col.32, lines 45-65). Thus, Ogg obviously includes a cryptographic function comprising a hash function and an exponentiation function. However, a secondary prior art is combined with Ogg to further clarify obviousness for binding and the claimed cryptographic function comprises a hash function and the hash function comprises an exponentiation function.

Graunke, et al. teaches an invention relates to digital content protection in computer systems and dynamically and securely distribute a private key over a network to access specific encrypted digital content (col.1, lines 7-12). (col.26, lines 49-51). Graunke discloses the present invention binds the integrity of a given application to its ability to perform some cryptographic operation using an asymmetric key pair in a manner that is tamper resistant. The goal is to prevent an unencrypted copy of digital content to be made. Graunke includes integrity verification kernels (IVKs), the use of an asymmetric key pair and a key compiler, and tamper resistance methods. It combines the cryptographic technologies of digital signatures and certificates with tamper resistant software to improve the integrity of the trusted player and a key module on the PC. Once these methods are used, this software is very difficult to modify without detection (col.4, lines 8-22). Thus, binding a secret to data obviously improves the integrity making it very difficult to modify without detection. Further, Graunke discloses asymmetric public key is passed to an Integrity Verification Kernel (IVK) generation (GEN) function whereby the IVK Gen function creates an IVK source code module that uses the

asymmetric public key as the root of trust (col.9, lines 25-30). The key compiler generates IVK source code for key module for calculating digital signatures and the source code which is output contains the "unrolled", optimized code for computing a cryptographic hash function followed by modular exponentiation (col.9, lines 32-41). Graunke discloses the manifest parser generator source code is static source code that includes the IVK's entry code, generator code, accumulator code, and other code for tamper detection. Supported services in this code include locating credentials and code using a registry, verification of object code prior to loading on disk and after loading in memory on the client, and validation of addresses in previously verified modules to provide secure linkage (col.9, lines 43-67).

Therefore, it would have been obvious for a person of ordinary skills in the art to combine the teachings of Ogg with Graunke to teach binding an initial secret to data indicative of an initial state of the system via a cryptographic function because to prevent an unencrypted copy of digital content to be made and to improve the integrity making it very difficult to modify without detection (Graunke – col.4, lines 8-22).

In addition, it would have been obvious for a person of ordinary skills in the art to combine the teachings of Ogg with Graunke to teach cryptographic function comprises a hash function and the hash function comprises an exponentiation function because to verify the data is validated and trusted for use in tamper detection of content (Graunke – col.9, lines 38-41 and col.10, lines 30-39).

As per claim 4: See Ogg on col.7, lines 43-44 and col.23, lines 20-34; discussing the method as claimed in claim 3, wherein the cryptographic function comprises a public/private key pair.

As per claim 5: See Ogg on col.22, lines 65-66; discussing the method as claimed in claim 3,

further comprising receiving the initial secret from a system administrator.

As per claim 8:

Ogg discloses a data processing system comprising:

a processor; a memory connected to the processor; and (col. 4, lines 62-63 and col.6, lines 23-26)

detection logic connected to the processor and the memory, the detection logic, in use: (col.7, lines 28-42)

providing an initial secret; (col.22, lines 48 and col.23, lines 46-48)

[binding] the initial secret (col.21, lines 7-14 and col.19, lines 6-18) to data indicative of an initial state of the system via a cryptographic function; (col.23, lines 57-67 and col.24, lines 45-52)

recording state changing administrative actions performed on the system in a log; (col.30, lines 44-47 and col.39, lines 50-52)

prior to performing each state changing administrative action, generating a new secret by performing the cryptographic function on a combination of data indicative of the administrative action and the previous secret (col.19, lines 44-48 and col.20, lines 39-52), and erasing the previous secret; (col.24, lines 58-67)

evolving the initial secret based on the log to produce an evolved secret; (col.33, lines14-19)

comparing the evolved secret with the new secret; (col.33, lines14-19)

determining that the system is uncorrupted if the comparison indicates a match between the evolved secret and the new secret; and (col.33, lines 21-25)

determining that the system in corrupted if the comparison indicate a mismatch between the evolved secret and the new secret; (col.33, lines 38-55)

[*wherein the cryptographic function comprises a one-way hash function and the hash function comprises an exponentiation function*]. (col.8, lines 10-40 and col.32, lines 45-65)

Ogg suggests private key used by modules to decrypt client secrets transmitted to the module during registration (col.23, lines 57-67 and col.33, lines 15-22), which seems to suggest a form of binding an initial secret to data indicative of an initial state of the system via a cryptographic function (col.19, lines 6-18 and col.21, lines 7-14). Ogg shows a table where the RSA engine is capable of performing the following modular arithmetic and exponentiation functions for high speed RSA encryption and including a signature (col.8, lines 10-40). Ogg discloses encryption and authentication of the key token (secret) format: DES, DSA, and RSA (col.18, lines 40-44 and col.22, lines 20-40). Ogg explains the HMAC is a digital signature created using a hash algorithm with an arbitrary message and the secret key (col.32, lines 45-65). Thus, Ogg obviously includes a cryptographic function comprising a hash function and an exponentiation function. However, a secondary prior art is combined with Ogg to further clarify obviousness for binding and the claimed cryptographic function comprises a hash function and the hash function comprises an exponentiation function.

Graunke, et al. teaches an invention relates to digital content protection in computer systems and dynamically and securely distribute a private key over a network to access specific encrypted digital content (col.1, lines 7-12). (col.26, lines 49-51). Graunke discloses the present invention binds the integrity of a given application to its ability to perform some cryptographic operation using an asymmetric key pair in a manner that is tamper resistant.

The goal is to prevent an unencrypted copy of digital content to be made. Graunke includes integrity verification kernels (IVKs), the use of an asymmetric key pair and a key compiler, and tamper resistance methods. It combines the cryptographic technologies of digital signatures and certificates with tamper resistant software to improve the integrity of the trusted player and a key module on the PC. Once these methods are used, this software is very difficult to modify without detection (col.4, lines 8-22). Thus, binding a secret to data obviously improves the integrity making it very difficult to modify without detection. Further, Graunke discloses asymmetric public key is passed to an Integrity Verification Kernel (IVK) generation (GEN) function whereby the IVK Gen function creates an IVK source code module that uses the asymmetric public key as the root of trust (col.9, lines 25-30). The key compiler generates IVK source code for key module for calculating digital signatures and the source code which is output contains the "unrolled", optimized code for computing a cryptographic hash function followed by modular exponentiation (col.9, lines 32-41). Graunke discloses the manifest parser generator source code is static source code that includes the IVK's entry code, generator code, accumulator code, and other code for tamper detection. Supported services in this code include locating credentials and code using a registry, verification of object code prior to loading on disk and after loading in memory on the client, and validation of addresses in previously verified modules to provide secure linkage (col.9, lines 43-67).

Therefore, it would have been obvious for a person of ordinary skills in the art to combine the teachings of Ogg with Graunke to teach binding an initial secret to data indicative of an initial state of the system via a cryptographic function because to prevent an unencrypted copy

of digital content to be made and to improve the integrity making it very difficult to modify without detection (Graunke – col.4, lines 8-22).

In addition, it would have been obvious for a person of ordinary skills in the art to combine the teachings of Ogg with Graunke to teach cryptographic function comprises a hash function and the hash function comprises an exponentiation function because to verify the data is validated and trusted for use in tamper detection of content (Graunke – col.9, lines 38-41 and col.10, lines 30-39).

As per claim 9: See Ogg on col.7, lines 43-44 and col.23, lines 20-34; discussing the system as claimed in claim 8, wherein the cryptographic function comprises a public/private key pair.

As per claim 10: See Ogg on col.7, lines 28-42***;*** discussing the system as claimed in claim 8, wherein the detection logic receives the initial secret from a system administrator.

As per claim 11: See Ogg on col. 4, lines 62-63 and col.6, lines 23-26***;*** discussing a computer program element comprising computer program code means which, when loaded in a processor of a computer system, configures the processor to perform a method as claimed in claim 3.

As per claim 13: See Ogg on col.7, lines 28-45 and col.24, lines 15-16; discussing a program storage device readable by machine, tangibly embodying a program of instructions executable by the machine to perform method steps for detecting an attack on a data processing system, said method steps comprising the steps of claim 3.

As per claim 14: See Ogg on col. 4, lines 62-63 and col.6, lines 23-26***;*** discussing a computer program product comprising a computer usable medium having computer readable program code means embodied therein for causing a data processing system, the computer readable program code means in said computer program product comprising computer readable program

code means for causing a computer to effect the functions of claim 8.

As per claim 15:

Ogg discloses a method for cryptographic entangling of state and administration in a data processing system, the method comprising:

initializing the system by generating an initial secret releasing (col.22, lines 48 and col.23, lines 46-48) *[binding]* data; (col.21, lines 7-14 and col.19, lines 6-18)

[binding the binding] data to the initial secret via a cryptographic function; (col.23, lines 57-67 and col.24, lines 45-52)

updating the initial secret in advance of an administrative action by computing a new secret; (col.4, lines 46-50 and col.19, lines 44-48 and col.20, lines 39-52)

erasing the initial secret together with any information from which the initial secret might be derived; (col.24, lines 58-67)

recording data indicative of the administrative action; (col.30, lines 44-47 and col.39, lines 50-52)

permitting execution of the administrative action; (col.14, lines 48-50 and col.24, lines 22-30)

offering a proof that the new secret corresponds to the initial secret as it has evolved according to a record of administrative actions, (col.33, lines 14-55)

[*wherein the cryptographic function comprises a one-way hash function and the hash function comprises an exponentiation function*]. (col.8, lines 10-40 and col.32, lines 45-65)

Ogg suggests private key used by modules to decrypt client secrets transmitted to the module during registration (col.23, lines 57-67 and col.33, lines 15-22), which seems to suggest

a form of binding an initial secret to data indicative of an initial state of the system via a cryptographic function (col.19, lines 6-18 and col.21, lines 7-14). Ogg shows a table where the RSA engine is capable of performing the following modular arithmetic and exponentiation functions for high speed RSA encryption and including a signature (col.8, lines 10-40). Ogg discloses encryption and authentication of the key token (secret) format: DES, DSA, and RSA (col.18, lines 40-44 and col.22, lines 20-40). Ogg explains the HMAC is a digital signature created using a hash algorithm with an arbitrary message and the secret key (col.32, lines 45-65). Thus, Ogg obviously includes a cryptographic function comprising a hash function and an exponentiation function. However, a secondary prior art is combined with Ogg to further clarify obviousness for binding and the claimed cryptographic function comprises a hash function and the hash function comprises an exponentiation function.

Graunke, et al. teaches an invention relates to digital content protection in computer systems and dynamically and securely distribute a private key over a network to access specific encrypted digital content (col.1, lines 7-12). (col.26, lines 49-51). Graunke discloses the present invention binds the integrity of a given application to its ability to perform some cryptographic operation using an asymmetric key pair in a manner that is tamper resistant. The goal is to prevent an unencrypted copy of digital content to be made. Graunke includes integrity verification kernels (IVKs), the use of an asymmetric key pair and a key compiler, and tamper resistance methods. It combines the cryptographic technologies of digital signatures and certificates with tamper resistant software to improve the integrity of the trusted player and a key module on the PC. Once these methods are used, this software is very difficult to modify without detection (col.4, lines 8-22). Thus, binding a secret to data obviously improves the

integrity making it very difficult to modify without detection. Further, Graunke discloses asymmetric public key is passed to an Integrity Verification Kernel (IVK) generation (GEN) function whereby the IVK Gen function creates an IVK source code module that uses the asymmetric public key as the root of trust (col.9, lines 25-30). The key compiler generates IVK source code for key module for calculating digital signatures and the source code which is output contains the "unrolled", optimized code for computing a cryptographic hash function followed by modular exponentiation (col.9, lines 32-41). Graunke discloses the manifest parser generator source code is static source code that includes the IVK's entry code, generator code, accumulator code, and other code for tamper detection. Supported services in this code include locating credentials and code using a registry, verification of object code prior to loading on disk and after loading in memory on the client, and validation of addresses in previously verified modules to provide secure linkage (col.9, lines 43-67).

Therefore, it would have been obvious for a person of ordinary skills in the art to combine the teachings of Ogg with Graunke to teach binding an initial secret to data indicative of an initial state of the system via a cryptographic function because to prevent an unencrypted copy of digital content to be made and to improve the integrity making it very difficult to modify without detection (Graunke – col.4, lines 8-22).

In addition, it would have been obvious for a person of ordinary skills in the art to combine the teachings of Ogg with Graunke to teach cryptographic function comprises a hash function and the hash function comprises an exponentiation function because to verify the data is validated and trusted for use in tamper detection of content (Graunke – col.9, lines 38-41 and col.10, lines 30-39).

As per claim 16: as rejected in claim 15; discussing a method as recited in claim 15, wherein the step of offering retrieves the initial secret via a request for entry of the initial secret by a system administrator, retrieving the record of administrative actions previous stored; and evolving a candidate secret for the initial secret based on the record of administrative actions retrieved; comparing the candidate secret with a current secret; if the candidate secret matches the current secret, reporting that the data processing system is still in an uncorrupted state, and if the candidate secret does not match the current secret, reporting that the data processing system is in a potentially compromised state.

As per claim 17: See Ogg on col.7, lines 28-45 and col.24, lines 15-16; discussing the method as recited in claim 15, further comprising permitting detection of any Trojan horse within the system.

As per claim 18: See Ogg on col.22, lines 48 and col.23, lines 46-48; discussing the method as recited in claim 15, wherein the initial secret is supplied via a secure communication channel.

As per claim 19: See Ogg on Scheidt on col.19, lines 18-33 and col.26, lines 49-51; discussing the method as recited in claim 15, wherein the binding data takes different forms depending on the data processing system, an application of the data processing system, and a trust mechanisms associated with communication of the initial secret.

As per claim 20: See Ogg on col.4, lines 46-50 and col.21, lines 4-26 and col.25, lines 20-55; discussing the method as recited in claim 15, wherein the administrative action is an action taken from a group of actions consisting of: updating of system executable code; updating of system libraries; installation of kernel modules; reading of files such as those used to store system states during rebooting operations; alteration of configuration files; alteration of system

run-level codes; writing to or reading from peripheral devices; and any combination of these actions.

As per claim 20: See Ogg on col.19, lines 6-18***;*** discussing a method as recited in claim 15, wherein the step of computing the new secret includes applying a one way function to a combination of a previous secret and data indicative of the administrative action.

Conclusion

5. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Leynna T. Truvan whose telephone number is (571) 272-3851. The examiner can normally be reached on Monday - Thursday (7:00 - 5:00PM).

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Kim Vu can be reached on (571) 272-3859. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/L. T. T./
Examiner, Art Unit 2135

/KIMYEN VU/

Supervisory Patent Examiner, Art Unit 2135